

Levenberg – Marquardt’s Algorithm used for PID Controller Parameters Optimization

Ahmed S. Abd El-Hamid, Ahmed H. Eissa, and ALy M. Radwan
National Research Center, Engineering Division, Dokki, Cairo, Egypt.
Email: ahmednrc64@gmail.com

Abstract: The determination of parameters of controllers is an important problem in automatic control systems. In this paper, the Levenberg Marquardt (LM) Algorithm is used to effectively solve this problem with reasonable computational effort. The Levenberg Marquardt (LM) Algorithm for optimization of three term (PID) controller parameters with dynamic model of pH neutralization process is presented. The main goal is to show the merits of Levenberg Marquardt algorithm optimization and to determine its suitability in the area of control systems. Lastly, the application of this approach to the calculation of the parameters of PID controller shows that the Levenberg Marquardt (LM) algorithm has a better dynamic performance of pH neutralization process.

Keywords: Parameters Optimization, PID controller, pH neutralization Process, Levenberg Marquardt (LM) algorithm

1. Introduction

In mathematics and computing, the Levenberg–Marquardt Algorithm (LMA), also known as the damped least-squares (DLS) method, is used to solve non-linear least squares problems. These minimization problems arise especially in least squares curve fitting. The LMA interpolates between the Gauss–Newton Algorithm (GNA) and the method of gradient descent.

The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach. The LMA is a very popular curve-fitting algorithm used in many software applications for solving generic curve-fitting problems. However, as for many fitting algorithms, the LMA finds only a local minimum, which is not necessarily the global minimum [1]. The primary application of the Levenberg–Marquardt algorithm is in the least squares curve fitting problem: given a set of m empirical datum pairs of independent and dependent variables, (x_i, y_i) optimize the parameters β of the model curve $f(x, \beta)$ so that the sum of the squares of the deviations becomes minimal.

$$S(\beta) = \sum_{i=1}^m [y_i - f(x_i, \beta)]^2$$

The Levenberg–Marquardt (LM) algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions [4, 6].

In the other hand, a common problem in control system design is establishing the appropriate value of controller gains. In general a low value of gain produces a slow system response, while high gain values can cause an excessively-oscillatory response with the possibility of instability. Somewhere between these extremes is a value of gain that produces the best system response. The essential function of a feedback control system is to reduce the error, $e(t)$ between any variable and its demanded value to zero as quickly as possible. Therefore, any criterion used to measure the quality of system response must take into account the variation of the error over the whole range of time. Four basic criteria are in common use:

$$\begin{aligned} \text{Integral of absolute error (IAE)} &= \int_0^\infty |e(t)| dt \\ \text{Integral of squared error (ISE)} &= \int_0^\infty \{e(t)\}^2 dt \\ \text{Integral of time multiplied by absolute error (ITAE)} &= \int_0^\infty t |e(t)| dt \\ \text{Integral of time multiplied by squared error (ITSE)} &= \int_0^\infty t \{e(t)\}^2 dt \end{aligned}$$

For any of the possible criteria, the best response corresponds to the minimum value of the chosen criterion. Note that in all cases it is either the absolute error or the squared error which is involved, straightforward integration of the error would produce zero result, even if the system response was a constant amplitude oscillation. IAE is often used where digital simulation of a system is being employed, but it is inapplicable for analytical work, because the absolute value of an error function is not generally analytic in form. This problem is overcome by the ISE criterion. The ITAE and ITSE have an additional time multiplier of the error function, which emphasizes long-duration errors, and therefore these criteria are most often applied in systems requiring a fast settling time [3, 5].

In this paper, the shape of the complete closed loop response, from time $t = 0$, until steady state has been reached, could be used for the formulation of a dynamic performance criterion. The simple criteria of this category are based on the entire response of the process and the integral of the Square Error (ISE) criterion used here, where

$$ISE = \int_0^\infty e^2(t) dt = \int_0^\infty \{Y_{sp}(t) - Y(t)\}^2 dt \quad (1)$$

Where $e(t) = Y_{sp}(t) - Y(t)$ is the deviation (error) of the response from the desired set point.

The ideal continuous time domain PID controller for a SISO process is expressed in the Laplace domain as follows:

$$G(s) = K_p + \frac{K_i}{s} + K_d s \quad (2)$$

With K_p = proportional gain, K_i = integral time constant and K_d = derivative time constant.

2. Mathematical model of pH neutralization process

Consider a pH neutralization process as shown in Fig. 1. The flow rates of acid, buffer, base and effluent streams are denoted by q_1, q_2, q_3 , and q_4 , respectively. Output of the process is the pH value of the effluent stream, and the flow rate of base stream, q_3 is the control input. A dynamic model is derived using the conservation laws and reactions equilibrium. The modeling assumptions include perfect mixing, constant volume of the neutralization tank (V), and complete solubility of the ions involved. The chemical reactions in the system are as follows [8]:

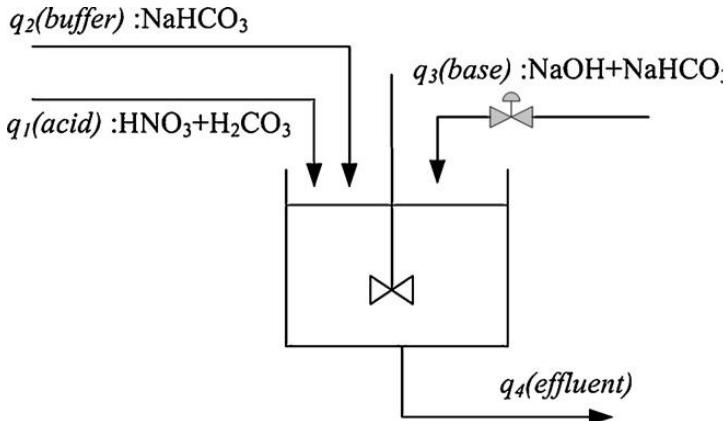
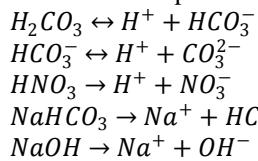


Fig. 1 pH neutralization process



The equilibrium constants for these reactions are:

$$Ka_1 = \frac{[HCO_3^-][H^+]}{[H_2CO_3]}$$

$$Ka_2 = \frac{[CO_3^{2-}][H^+]}{[HCO_3^-]}$$

$$Kw = [H^+] + [OH^-] + [CO_3^{2-}]$$

The chemical equilibrium equations are modeled using the reaction invariant concept. For this system, concentrations of reaction invariants are defined as:

$$\begin{aligned} x_1 &= [NO_3^-] \\ x_2 &= [Na^+] \\ x_3 &= [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}] \end{aligned}$$

Denoting, $y = pH$ the ions neutrality balance in the tank results the following static equation:

$$\begin{aligned} h(x, y) &= -x_1 + x_2 + x_3 c_{x3} + 10^{-y} - \\ 10^{y-PK_w} &= 0 \end{aligned} \quad (3)$$

$$c_{x3} = \frac{2 + 10^{PK_2-y}}{1 + 10^{PK_2-y} + 10^{PK_1+PK_2-2y}}$$

$$PK_1 = -\log_{10} Ka_1$$

$$PK_2 = -\log_{10} Ka_2$$

The dynamic equations are given by:

$$\frac{dx_1}{dt} = \frac{q_1}{V} (w_{11} - x_1) + \frac{q_2}{V} (w_{21} - x_1) + \frac{q_3}{V} (\alpha_1 - x_1) \quad (4)$$

$$\frac{dx_2}{dt} = \frac{q_1}{V} (w_{12} - x_2) + \frac{q_2}{V} (w_{22} - x_2) + \frac{q_3}{V} (\alpha_2 - x_2) \quad (5)$$

$$\frac{dx_3}{dt} = \frac{q_1}{V} (w_{13} - x_3) + \frac{q_2}{V} (w_{23} - x_3) + \frac{q_3}{V} (\alpha_3 - x_3) \quad (6)$$

Where:

V : Volume of the mixing tank, ml

Kw : Dissociation constant of water, 10^{-14}

Ka_i : ith dissociation constant of acid

w_i : Concentration of the ith species in the process stream, mol/l

w_{1i} : Concentration of the ith species in the acid stream, mol/l

w_{2i} : Concentration of the ith species in the buffer stream, mol/l

q_i : Flow rate of acid, buffer and base stream in simulation, ml/s

α_i : Concentration of the ith species in the titrating stream, mol/l

x_i : Reaction invariant of ith species, mol/l

y : Process variable, pH

u : Flow rate of the titrating stream, ml/min or ml/s

3. Simulation Results

The closed loop control system was solved using Levenberg–Marquardt's optimization approach with sampling time of 0.001 s. The simulation method combines SIMULINK module for pH neutralization model and M-file for LMA approach. A list of M-file programs used in the paper is provided in Appendix 1 and 2. Figure (1 and 2) show the responses of the pH neutralization obtained with change in pH set point. The optimal gains of PID controller are calculated to minimize the error function which described in equation (2). Also the values of gains of PID controller are plotted. The response of pH value tends to set point value with minimum steady state error and the values of gains tend to minimum values once the error reaches to zero value. Figure (3) show the change in set point from pH = 6 to 7, noticed that the spike value of manipulated variable (q_3) and the values of gains (K_p and K_i) tend to increasing while the gain (K_d) tends to decreasing. In other hand, figure (4) show the change in set point from 7 to 6, the value of manipulated variable (q_3) and the value of gains (K_p , K_i and K_d) tend to decreasing.

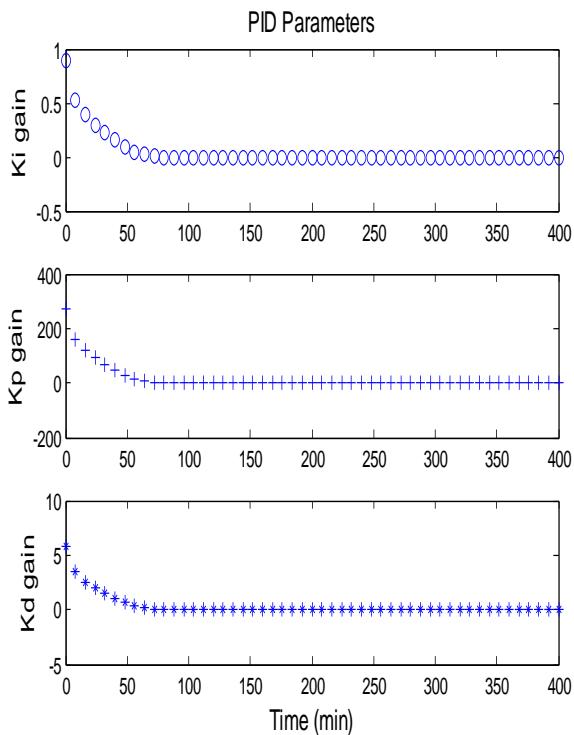
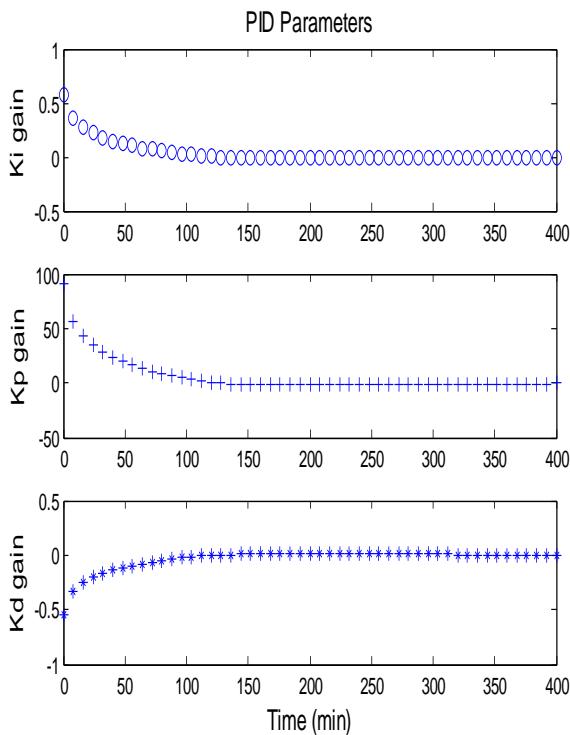
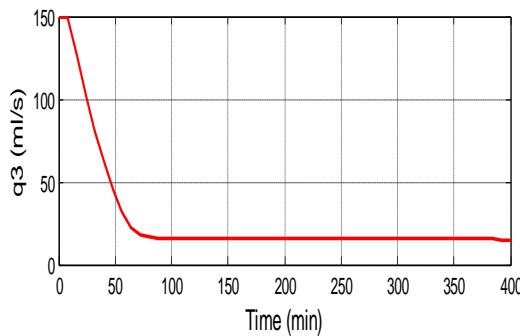
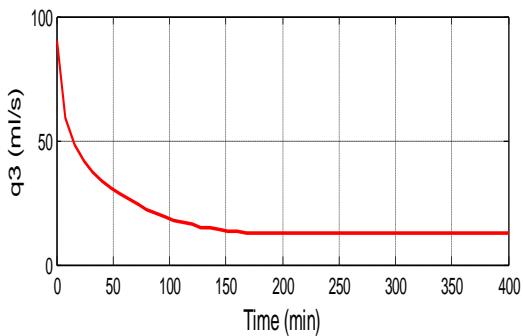
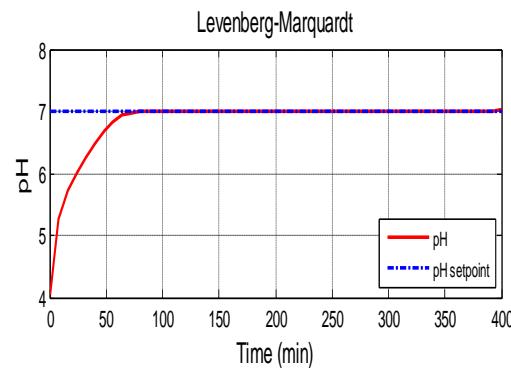
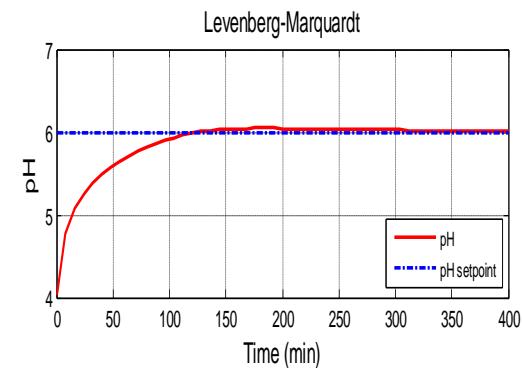


Fig. 1. Simulation results of PID controller and values of parameters

Fig. 2. Simulation results of PID controller and values of parameters

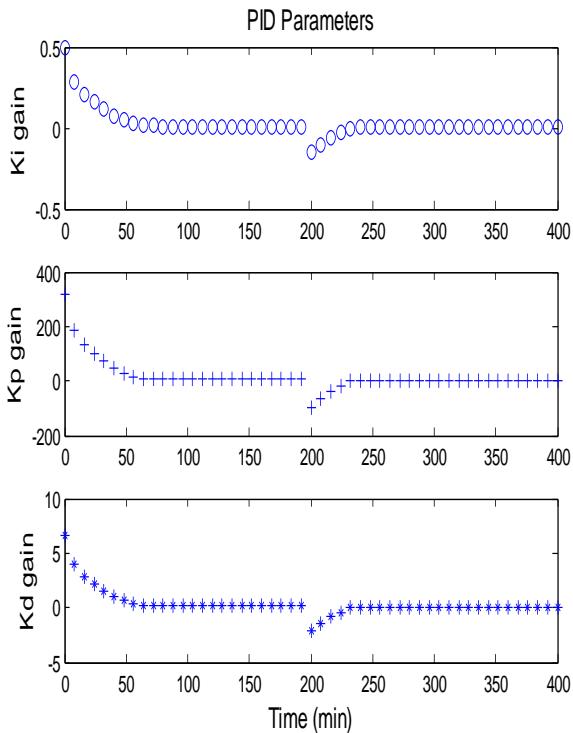
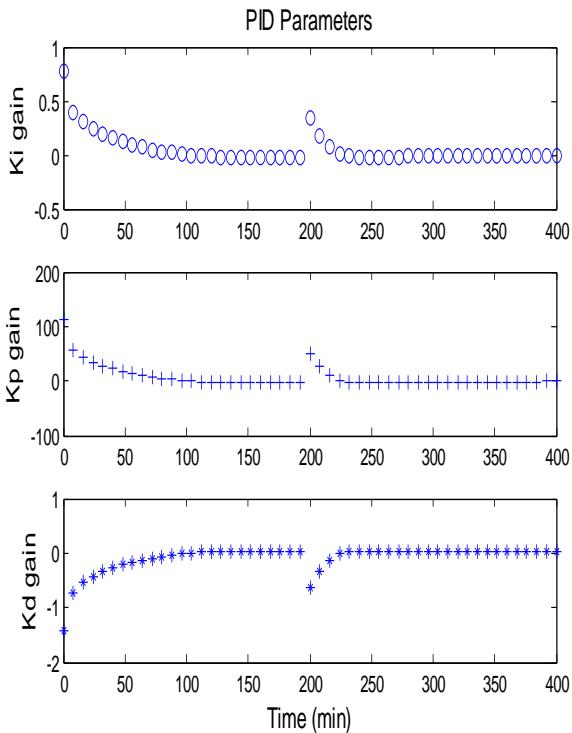
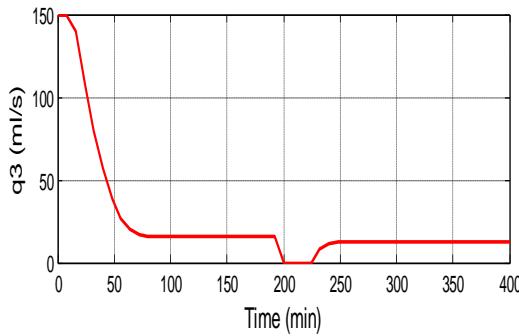
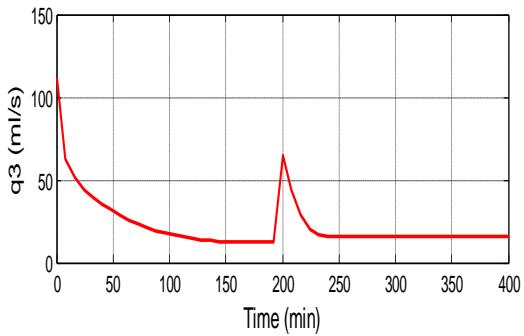
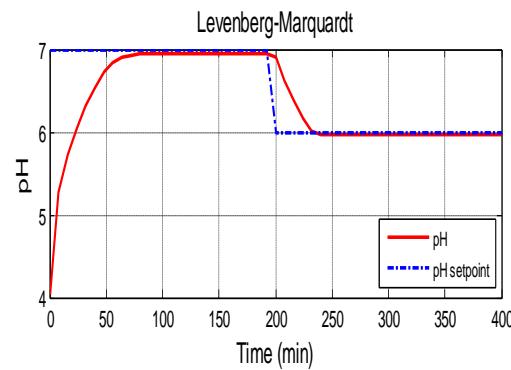
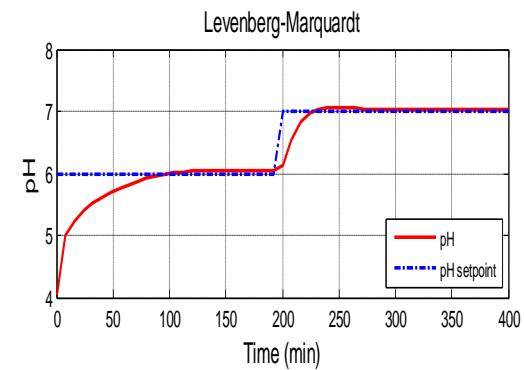


Fig. 3. Simulation results of PID controller and values of parameters

Fig. 4. Simulation results of PID controller and values of parameters

CONCLUSION

The paper presents an application of the Levenberg-Marquardt Algorithm (LMA) to optimization of parameters (K_p , K_i , K_d) of the PID controller structure according to minimum of integral square error. The simulated results were obtained of parameters by means of computer program implemented in Matlab software. As an example, the optimization of parameters of PID controllers with reference to a pH Neutralization process was presented.

Reference

- i. "Fuzzy-neural predictive control using Levenberg-Marquardt optimization approach"
- ii. Ayman A. Aly "PID Parameters Optimization Using Genetic Algorithm Technique for Electrohydraulic Servo Control System" *Intelligent Control and Automation*, 2011, 2, 69-76
- iii. P. Cominos and N. Munro, "PID Controllers: Recent Tuning Methods and Design to Specification," *IEE Proceedings Control Theory & Applications*, Vol. 149, No. 1, January 2002.
- iv. W. Wang, J. T. Zhang and T. Y. A. Chai, "Survey of Advanced PID Parameter Tuning Methods," *Acta Automatica*, Vol. 26, No. 3, 2000, pp. 347-355.
- v. Kamran Turkoglu, Ugur Ozdemir, Melike Nikbay, and Elbrouss M. Jafarov "PID Parameter Optimization of an UAV Longitudinal Flight Control System" *World Academy of Science, Engineering and Technology* 21 2008
- vi. B. Kada, Y. Ghazzawi "Robust PID Controller Design for an UAV Flight Control System" *WCECS 2011*, October 19-21, 2011, San Francisco, USA
- vii. Mendis. B. S. U., Gedeon. T. D, Kóczy. L. T (2006). "Learning Generalized Weighted Relevance Aggregation Operators Using Levenberg-Marquardt," *Proc. of the Sixth IEEE International Conference on Hybrid Intelligent Systems*.
- viii. Meng. X, Song. B (2007). "Fast Genetic Algorithms Used for PID Parameter Optimization." *Proceedings of IEEE International Conference on Automatic and Logistic*, pp. 2144-2148.

Appendix (1)

```
%Determination of PID controller parameters using "Levenberg-Marquardt %Algorithm"
%Step 1: Write an M-file tracklsq.m.
function [Kp,Ki,Kd] = phpidL2015k
ph2015 % Load the simulink model
k0 = [0 0 0]; % Set initial values of parameters
%%%%%%%%%%%%%%%
%%%Create or edit optimization options structure
options = optimset('Algorithm','levenberg-
marquardt','Display','iter',...
'TolX',1e-5,'TolFun',1e-9,'TolCon',1e-6);
k = lsqnonlin(@tracklsq, k0, [], [], options);
Kp = k(1); Ki = k(2); Kd = k(3);
function F = tracklsq(k)
Kp = k(1);
Ki = k(2);
Kd = k(3);
```

```
%%%%%%%%%%%%%%
% Choose solver and set model workspace to this function
%Step 2: Invoke optimization routine.
simopt = simset('solver','ode5','SrcWorkspace','Current');
[t,x,y1,y2,y3] = sim('ph2015',[0 400],simopt);

%%%%%%%%%%%%%
%F = set point - actual value; % Compute error value
F = y2-y1;
end
%
% Put variables back in the base workspace
Kp = k(1)
Ki = k(2)
Kd = k(3)
%
%plot(tout,yout,'r',tout,yset,'-b');
% Set axes and labels.
%axis([0 30 -1.6 1.6]); xlabel('Time'); ylabel('Amplitude');
clf reset
subplot(2,1,1),plot(t,y1,'r',t,y2,'-b','LineWidth',2);
title('levenberg-marquardt')
legend({'pH','pH setpoint'},'Location','SE','FontSize',8')
xlabel('Time (min)','FontSize',12')
ylabel('pH','FontSize',12)
grid on
subplot(2,1,2),plot(t,y3,'r','LineWidth',2);
xlabel('Time (min)','FontSize',12)
ylabel('q3 (ml/s)','FontSize',12)
grid on
end
```

Appendix (2)

```
% Read Mat. Files of PID gains
% ki(First number, second number)first number points to raw number
% second number points to col. number
%
load ki.mat; %# assume this contains a matrix called ki
load kp.mat
load kd.mat
for i=1:1:51 %# numbers of columns
x(i) = ki(1,i); %# numbers from all rows, column 1, into X
x1(i)=kp(1,i);
x2(i)=kd(1,i);
y(i)=ki(2,i); %# numbers from all rows, column 2, into Y
y1(i)= kp(2,i);
y2(i)=kd(2,i);

subplot(3,1,1),(plot(x,y,'o'));
title('PID Parameters','FontSize',12')
ylabel('Ki','FontSize',12)
subplot(3,1,2),(plot(x1,y1,'+'));
ylabel('Kp','FontSize',12)
subplot(3,1,3),(plot(x2,y2,'*'));
xlabel('Time (min)','FontSize',12')
ylabel('Kd','FontSize',12)

end
```